土木建筑工程信息技术
Journal of Information Technology in Civil Engineering and Architecture



# 基于事件监听机制的建筑信息模型的版本管理方案

桂 宁 张 新 邓小翔 宋瑾玉

(浙江理工大学信息学院,杭州 310018)

【摘 要】建筑信息模型(BIM),以其强大的建筑数字化能力,已经在建筑生命周期的越来越多的领域和流程得到应用。在建筑设计过程中,BIM模型需要经过多次的修改才能形成一份完善的项目设计方案,这些修改信息可以为方案的优化设计提供借鉴,但是Revit不能对这些中间状态的信息提供详细地记录和有效地跟踪。为了实现对模型修改过程的全程跟踪和提高BIM模型的修改效率,提出了一个基于事件监听机制的建筑信息模型的版本管理方案,研究表明利用这个版本管理方案可以有效地实现对模型各种版本的修改信息的跟踪,方便了建筑信息模型的管理;同时,版本管理方案采取合并操作信息的机制,对同一对象的冗余操作进行清除,缩短了模型从初始状态到目标状态的执行时间,起到比较明显的加速作用。

【关键词】建筑信息模型;事件监听机制;版本管理;操作追踪
【中图分类号】TU17 【文献标识码】A 【文章编号】1674-7461(2017)02-0092-06
【DOI】10.16670/j.cnki.cn11-5823/tu.2017.02.16



## 引言

建筑信息模型[1](BIM, Building Information Modeling)是以建筑项目整个生命周期的数据信息为基础,通过数字化来模拟真实建筑物的信息而建立的多维的建筑模型。北美建筑行业[2]有60%的机构使用BIM技术;欧洲、日本和香港地区,BIM技术已经广泛应用于各类房地产的开发[3]。

BIM项目设计是一个各专业同步设计[4,5]的过程,不是从二维图纸到三维模型的简单转变,能够满足版本管理方案设计的需求[6]。然而,BIM项目设计过程中容易出现多版本[7]的现象。首先,BIM设计使处于不同时间、地点、领域的专家能够参与同一项目的设计,决定了设计时必然会产生多个版本。其次,项目设计师思路的调整可能出现多个版本。另外,用户等项目参与方对设计意见的反馈也会引发多版本的出现[8]。而这些中间状态的版本是在大量的分析和实践中获取的,对目标方案的获取起到巨大的指导作用:第一,无论是横向的相互协商还是纵向的设计反复,都不会很快获得目标方案,存储这些中间版本便于进行比较和回顾;第二,设计过程中产生的版本信息,蕴含着丰富的经验和启示,对中间版本进行管理,有助于对专家系统的设计知识库进行扩充,实现智能化设计。因此,对中间版本的信息进行有效地管理是一项亟待解决的课题。

目前,已开发了众多的BIM平台,如XBIM、IFC Web Server、IFC Viewer等。XBIM主要实现IFC和模型数据的简单统计;IFC Web Server主要实现IFC文件与2D、3D模型转换;IFC Viewer主要实现了模型的共享和展示功能。但是,这些平台立足于建筑模型图形展示功能的实现,不支持模型数据的版本控制,缺乏对数据信息有效的版本管理。BIM Server[9]是目前应用较为广泛的BIM服务器平台,提供了版本比较的功能来进行BIM模型的管理,但

【基金项目】 教育部——欧特克公司专业综合子课题改革项目(项目编号:joz201401)
【作者简介】 桂宁(1977-),男,比利时鲁汶大学博士后,硕士生导师。主要从事基于BIM的新能源建筑仿真、光伏建筑一体化辅助设计和大数据方面的研究。

不能对模型提供完善的编辑功能。

Revit 是一款比较常用的商用 BIM 设计软件，提供了强大的模型编辑功能，内部有自带的版本管理方案：通过本地模型与服务器或者本地电脑上的中心模型进行关联来更新中心模型，其他开发人员可以依据中心模型获取最新的建筑模型。但是，分析发现，Revit 内部的历史记录包含的信息有限（包含日期、时间标记、修改者和注释），不具备向所有的模型状态进行回溯和前进的条件。Autodesk Vault 虽然支持版本控制，但项目参与方有存储和传输数据的需求，会耗费大量的存储和解析开销，需要提供额外的服务器等设备，增加了项目设计的成本。

本文提出了一种基于事件监听机制的建筑信息模型的版本管理方案，在 Autodesk Revit 平台上实现了一种本地化、轻量级的版本管理：重用了 Revit 内部的事件监听机制，实时、动态地监听用户对 BIM 模型的修改；设计修正模块来支持用户对 BIM 模型进行添加、删除和参数修改等操作；获取监听信息并以可扩展语言 Xml 的形式记录存储以及调用；在面向对象操作的基础上，实现对模型的修改版本的追踪；进而优化操作信息的处理方式，通过 BIM 模型与存储区信息的结合实现模型版本的更新，提升 BIM 项目的设计效率。



## 1 模型修正与事件监听机制

BIM 模型是由大量的对象组成的，Revit 提供可以重用的事件机制、SDK 和 API 来支持对象的修改。因而可以通过对象的修改实现模型的修改。图 1 是一个 BIM 模型的实例：

在 Revit 平台上，建立小屋模型。而操作信息的生成、监听和获取是实现版本管理的关键，本文通过设计修正模块和事件监听机制来实现操作信息的前期处理。以 BIM 模型为基础，进而实现对象的增加、删除和参数修改等操作以及监听。

### 1.1 模型修正

BIM 模型可以存储一幢建筑完整生命周期内的数据信息[10, 11]，占据空间可达数百兆；建筑项目从设计到完成也会经历较长的时间。大量的数据信息、长时间的建筑施工以及众多项目关联方的参与，增加了 BIM 模型调用、设置和修改的频率。

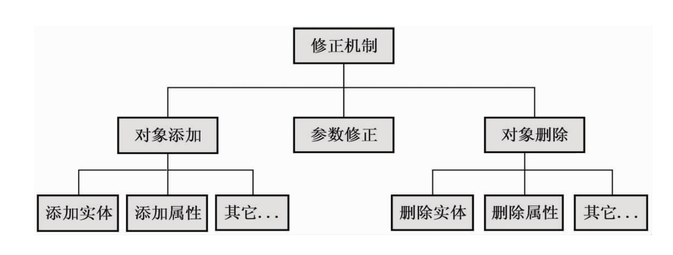



图 2 修正模块

图 2 是修正模块的设计框图。修正模块需要实现模型的修改操作：可以在模型中添加窗子、柱子等建筑组件和电气组件；可以对模型中存在的对象进行删除操作；也可以对各种组件的参数进行修改等等。

### 1.2 事件监听机制

BIM 模型修改后，实现版本管理需要获取用户的操作信息。图 3 是设计的事件监听机制运行框图：

随着修改的进行，Document 的变化会触发大量的事件（如 DocumentOpened、DocumentClosed、FileImported、FileExported、DocumentChanged 等），本方案通过重用 Revit 内部的事件机制来实现监听。但后续模块不需要涉及到所有的事件，需要与事件过滤器结合。模型的变化作用于事件引擎，过滤获取

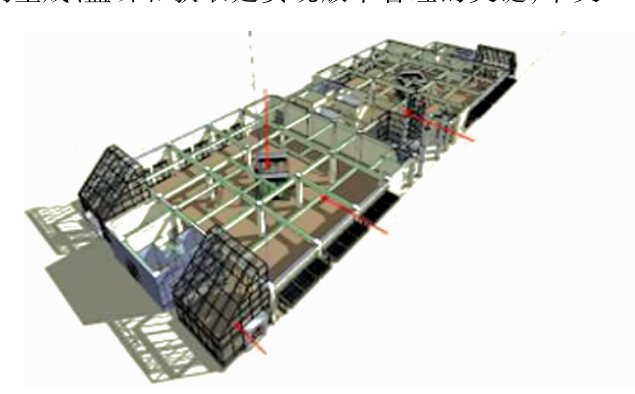
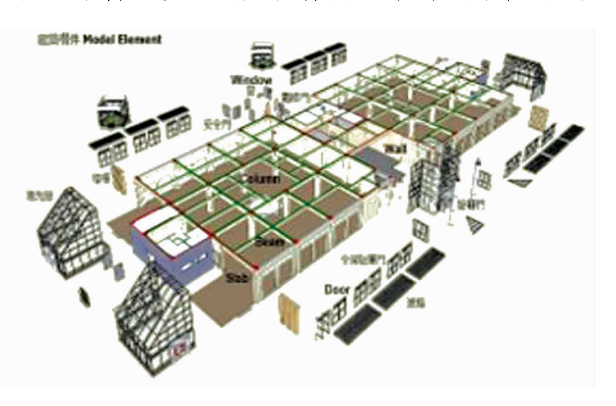



图 1 BIM 模型与爆炸图

DocumentChanged 事件。将监听的操作信息导出 Revit 软件,以 Xml 文件的形式存储。

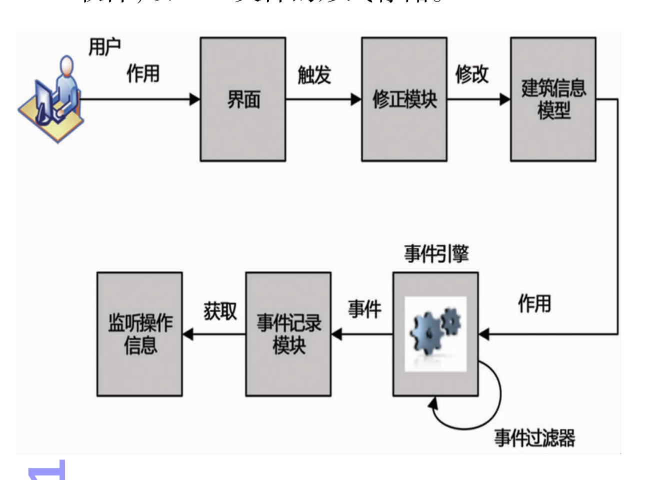



图 3 事件监听机制运行

## 2 版本管理方案

在 BIM 项目设计过程中,由于设计人员资历、经验以及项目参与方等多种因素的影响,会出现多版本的状态。而一个优秀的设计方案通常是在综合分析和比较多个不同的设计版本后才能够获取,这决定了版本管理方案需要兼顾[12, 13]两方面的功能需求:既可以存储最终的设计版本,同时又能保留设计过程的中间状态的版本[14, 15],便于获取前期方案的经验和启示更好地进行设计,本方案通过修改信息与初始模型的结合来实现。

### 2.1 操作信息的存储

用户的修改信息,既可以作为修改版本提交用户,又可以继续完善。而监听机制获取的操作信息需要进行存储,便于版本管理时调用。为了满足功能需求并处理好不同的业务逻辑,将位于项目内部的存储区设计为三部分:操作库、版本库、快照库,三个库是相互依赖的整体[16-19]。

BIM 模型的修改过程如图 4 所示。

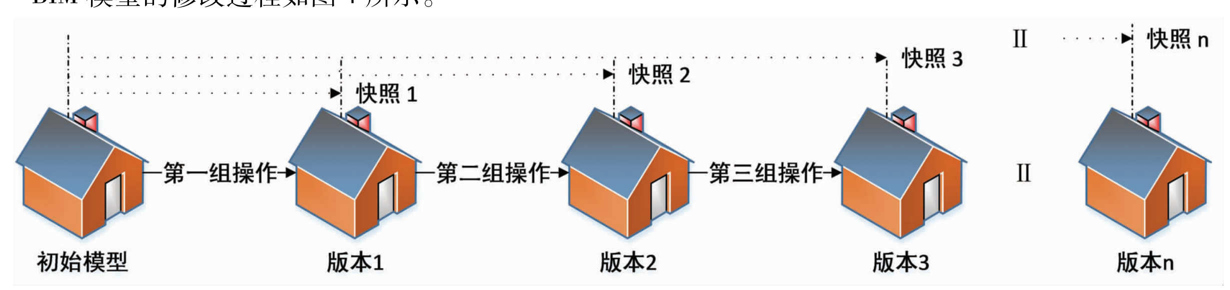



图 4 模型修改

BIM 模型修改过程中,每一组模型的修改操作,在成功提交后视为一个版本信息;多次提交就有多个版本信息。BIM 模型在操作信息的作用下实现了更新。

存储区结构设计如下:

操作库:记录 BIM 模型从初始状态到版本提交时所做的修改;

版本库:记录成功提交的版本信息,并保存版本库的属性信息(如版本总数、当前版本号等信息);

快照库:内部保存的是一组操作集。将快照信息作用于初始模型,可以获取相应版本的模型状态;同时包含快照的一些属性信息。默认情况下,快照库记录了从初始模型到当前版本 n(可取 1, 2, 3, 4……)的操作信息;用户设定目标版本后,依据版本记录生成快照信息,为版本回溯和前进提供前提条件。

工作区与存储区之间的关系如图 5 所示。

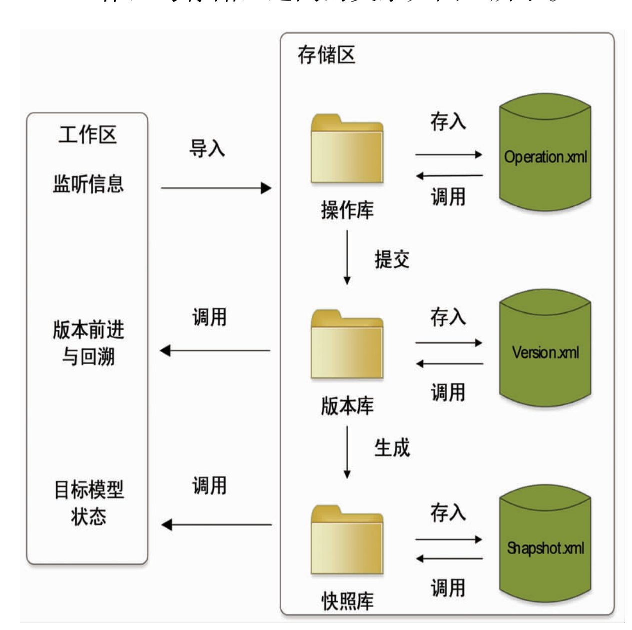



图 5 工作区与存储区关系

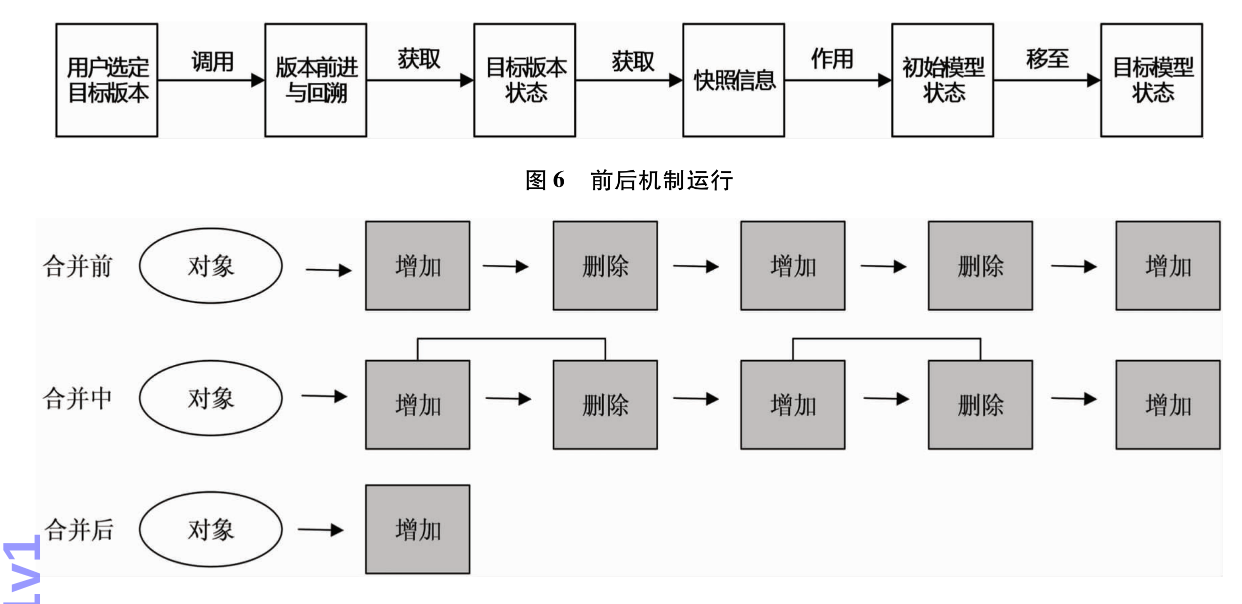



图 6　前后机制运行

图 7　对象增加与删除的合并设计



用户在工作区对 BIM 模型进行修改或版本的前进、回溯；于存储区将监听的信息保存和提交，生成版本库信息和快照库信息。

模型修改过程时，Revit 通过监听机制获取目标事件，监听信息保存在操作库中；用户可以向版本库提交操作信息；快照库的信息则依据版本库中的信息生成，默认情况下，保存的是从初始模型到当前所处版本的快照信息。基于初始 BIM 模型，与快照信息结合向目标模型状态更新。

### 2.2　操作追踪

BIM 模型修改后，监听的操作信息通过提交就会获取一个版本信息。用户将每个阶段的修改信息都提交到版本库中，就能实现模型整个修改过程的记录。而版本库的记录在退出项目后依然存在，版本管理的效果不会受到影响。

### 2.3　前后机制

BIM 模型的修改信息存储于版本库中，通过设定目标版本，依据前后机制中设计的方法和版本库中的信息，快照库中会生成模型起始状态到目标版本状态的快照信息。图 6 是详细的前进和回溯运行图。

监听机制获取的一个重要属性就是操作类型(Deleted，Added，Modified)。基于上述属性，开发前进回溯功能模块，实现 BIM 项目设计的版本管理。

版本管理过程中，获取的操作信息存在可以合并的状态。而前后机制的时间开销取决于版本库实际的操作信息的数量，实现合并冗余操作能够压缩前进或回溯的时间，进而提升版本管理的效率。

本文研究的合并状态如下：

1)参数修改时，对象某一参数的多次修改，可以合并；

2)同一对象执行添加或删除操作可以合并，图 7 为该情况下的一种合并情形。

然而，合并需要具备一定的条件。参数修改时，需考虑合并的操作信息是否位于同一个对象的相同参数；对象添加(或删除)时，操作对象是否位于同一个位置，依据操作类型合并可逆操作信息；版本中合并的操作信息是否相连(图 7 为可逆操作相连状态)等情况。制定统一的合并规则，进行合并的开发。

## 3　功能测试和性能分析

本管理方案已在 Autodesk 公司的 BIM 设计软件 Revit 2015 上开发并实现。通过重用 Revit 2015 内部事件机制，实现了 BIM 模型修改过程的跟踪，开发了修正模块，设计了事件监听机制，通过 DocumentChanged 事件监听 BIM 模型的修改，获取监听信息并导出 Revit 软件。以用户选定的版本为目标状态，应用设计的前进(或回溯)模块向目标状态更新。

图 8 是版本管理方案的系统实现。以提交的 3 个版本为例，介绍了 BIM 模型的修改；监听信息的

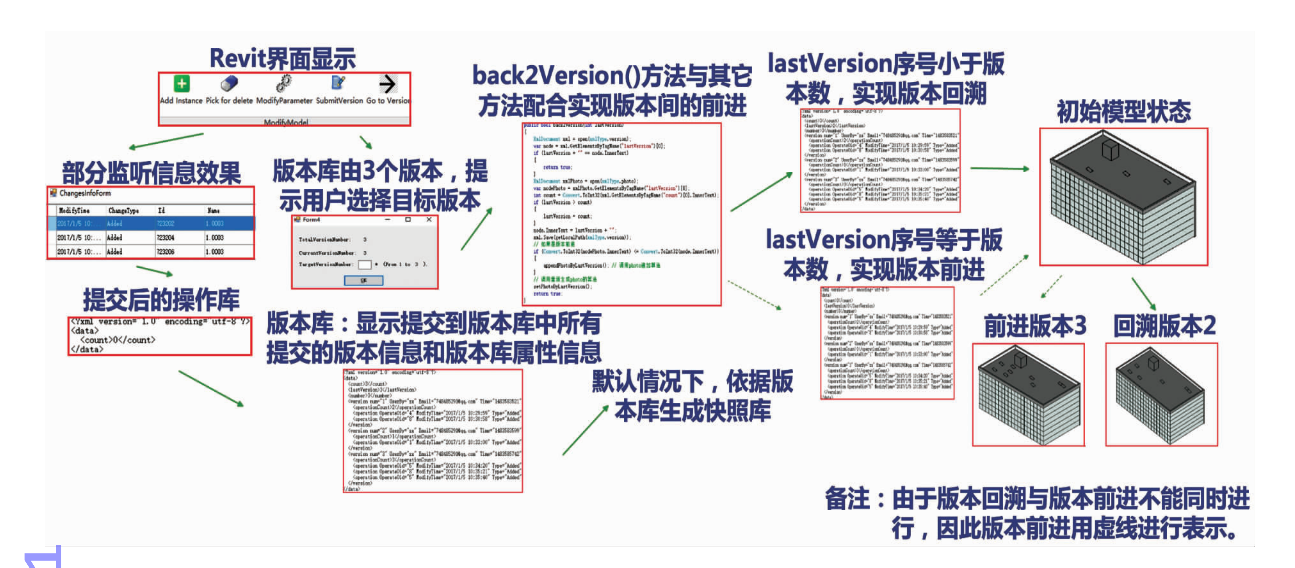



图 8　版本管理方案的实现



获取、导出与提交；存储区的展示；版本前进和回溯的运行流程；比较了不同情况下版本库的变化情况；获取了更新后的 BIM 模型。运行效果表明：无论是版本前进或回溯，BIM 模型均可以向目标版本状态更新。

为了分析合并的效果，搭建测试环境对模型的参数修改、对象的添加与删除操作的时间分别进行测试。环境构建如下：硬件配置是内存为 8G、处理器为 Intel(R) Xeon(R) CPU E3-1226v3 @ 3.30GHz 和具有 64 位 Windows10 操作系统的惠普工作站，软件配置为 VS 2015 专业版和 Revit 2015 简体中文版。随机地修改模型，测试起始模型状态到目标模型状态的生成时间，获取的数据如表 1 所示：

表 1　模型修改的效果

| 执行时间(ms) / 测试(次) | 合并前 | 合并后 |
|---|---|---|
| 18 | 391.6 | 112.2 |
| 60 | 1 149.5 | 126.6 |
| 100 | 1 937.5 | 127.4 |

分析发现：合并之前，系统执行时间达数百毫秒，随着操作次数的增加，时间逐步增加；合并后，操作信息数量减少，较大幅度地缩短了模型从初始状态到目标状态的时间，验证了操作合并的效果。而一个大的 BIM 项目需要多次的修改，在数千次的修改操作中，与不合并的设计方式相比，借助于合并思路设计的前后机制能快速的前进(或回溯)到目标版本状态，为建筑设计提供了较大的便利。

## 4　结语

本文在事件监听机制的基础上，将操作信息实时地提交，获取了用户对 BIM 模型的详细修改记录，实现了模型修改的操作追踪和版本管理。性能测试发现：与不合并的设计方式相比，方案如果采取合并操作的设计思路，能有效地节约从初始状态到目标状态的时间开销，提高了版本管理的效率。

文中的版本管理方案具有如下特点：

1)修正模块能实现模型的基础性的修改；

2)重用 Revit SDK，生成事件机制，监听 BIM 模型地修改；

3)存储区分为操作库、版本库、快照库三个部分；三个库互相关联，形成一个符合逻辑、功能清晰的整体；

4)针对版本状态的变化，设置针对性的快照生成方式。

本方案能实现 BIM 模型状态的前进和回溯，可以帮助工程项目人员有效地进行 BIM 项目设计，是一次理论联系实际的有益探索。但在修正机制的设计方面，还仅限于一些基础性的修改操作，功能有待完善。

## 参考文献

[1] EASTMAN C. What is BIM[J]. BIM resources @ Georgia Tech, 2009.

[2] 黄亚斌. BIM 技术在设计中的应用实现[J]. 土木建筑

工程信息技术，2010，04）：71-8.
[3] 宝琦. 基于BIM的工业建筑协同设计[J]. 工业建筑，2010，S1：84-6.
[4] LIU Y, VAN NEDERVEEN S, HERTOGH M. Understanding effects of BIM on collaborative design and construction: An empirical study in China[J]. International Journal of Project Management, 2016.
[5] ZADA J, JALAL A. Versioning of IFC-based information models for collaborative design[D]. University of Nottingham, 2016.
[6] 李建波，穆华倩. 基于REVIT的建筑电气BIM协同设计分析[J]. 智能建筑电气技术，2015，01：8-11.
[7] 曹健，张友良，赵海燕. 协同设计的版本管理[J]. 计算机集成制造系统-CIMS，1998，06：17-21.
[8] 潘翔，王云鹏，郭学旭，等. 协同设计中的版本管理解决方案[J]. 航空制造技术，2004，01：66-9.
[9] JIANG Y, MING J, WU D, et al. BIM server requirements to support the energy efficient building lifecycle[C]. proceedings of the Proc 2012 ASCE international conference on computing in civil engineering, 2012.
[10] 刘照球，李云贵. 建筑信息模型的发展及其在设计中的应用[J]. 建筑科学，2009，01：96-9 + 108.
[11] 秦军. 建筑设计阶段的BIM应用[J]. 建筑技艺，2011，Z1：160-3.
[12] 宋政君，黄克正，杨志宏，等. 协同概念设计中的版本管理[J]. 机电一体化，2004，02：93-6.
[13] 殷洁. 建筑设计协同平台研究设计[D]. 上海交通大学，2015.
[14] 方婉蓉. 基于BIM技术的建筑结构协同设计研究[D]. 武汉科技大学，2013.
[15] 黄琢华. 基于BIM的分布式协同设计平台底层框架研究[C]. 信息化推动工程建设工业化——第四届工程建设计算机应用创新论坛，上海，2013.
[16] 王潜平，林宗楷，郭玉钗. 支持协同设计的工程数据库版本管理[J]. 软件学报，1996，11：52.
[17] 刘国军. 产品协同设计过程的版本管理研究[D]. 太原科技大学，2011.
[18] 刘玲. 支持需求跟踪的版本控制技术的研究[D]. 电子科技大学，2008.
[19] 陈虔，刘慧，冯允成，等. 分布协同开发的软件库版本控制[J]. 系统工程理论与实践，2000，9：64-8 + 73.



# Version Management Scheme of Building Information Modeling Based on Event Monitoring Mechanism

Gui Ning, Zhang Xin, Deng Xiaoxiang, Song Jinyu

(*School of Information Science and Technology, Zhejiang Sci-Tech. University, Hangzhou* 310018, *China*)

**Abstract**: The Building Information Modeling (BIM), possessing powerful architecture digitizing capability, has been increasingly applied in construction fields and processes during the life cycle. In the architecture designing process, the building information model should be modified several times before perfectly forming a complete project design scheme, and those modification information can provide references for the optimization of the design scheme. However, Revit can not provide detailed recordings or eligible traces of those intermediate state information. In order to achieve all-the-way tracking and improve efficiency of the model modifications, this paper presents a version management scheme of building information modeling based on event monitoring mechanism. Research shows that such a version management scheme can effectively track modification information of various model versions, which facilitates the management of building information modeling. Meanwhile, the version management scheme adopts the mechanism of merging operation information to clear the redundant operation of the same object, which shortens the execution time of the model from the initial state to the target state, and plays a significant role in acceleration.

**Key Words**: Building Information Modeling (BIM); Event Monitoring Mechanism; Version Management; Operational Tracking